

# L4 微内核操作系统及其应用技术研究

李 鹏<sup>1</sup>, 张 凡<sup>2</sup>

(1. 中航工业西安航空计算技术研究所, 陕西 西安 710068;  
2. 西北工业大学 计算机学院, 陕西 西安 710072)

**摘 要:** 嵌入式系统对其操作系统的可靠性、实时性和安全性有很高的要求, 微内核操作系统技术能够很好地匹配该需求, 因此研究微内核在实际系统中的应用和开发方法对嵌入式应用领域的发展有着重要意义。文中对 L4 微内核的结构及其实现原理进行了深入的分析。在此基础上, 给出了基于 L4 微内核进行应用开发的一般步骤。同时依据此步骤, 实现了一个检测和打印嵌入式系统硬件参数的应用实例。通过实例实现, 认识到在嵌入式系统的应用开发中, 熟悉内核的功能接口将有助于更好地进行应用开发。

**关键词:** 微内核; 嵌入式操作系统; 进程间通信; L4

**中图分类号:** TP316.2

**文献标识码:** A

**文章编号:** 1673-629X(2014)04-0029-04

doi:10.3969/j.issn.1673-629X.2014.04.007

## Research on L4 Microkernel Operating System and Its Application Technology

LI Peng<sup>1</sup>, ZHANG Fan<sup>2</sup>

(1. Xi'an Aeronautics Computing Technique Research Institute of AVIC, Xi'an 710068, China;  
2. School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** The embedded systems have high requirements for the operating system's reliability, real time, and safety. The microkernel operating system fits the requirements well. Therefore the means of applications of microkernel will play an important role in the embedded applications. In this paper, analyze the structure and the principle of L4 microkernel in detail. On this basis, describe the general approaches to develop applications based on L4 microkernel. At the same time, an application example of detecting and printing embedded system parameters is realized. And through the realization of the instance, recognize that it is important for developers to be familiar with the function interfaces of the microkernel in the application development of embedded system.

**Key words:** microkernel; embedded operating system; Inter-Process Communication (IPC); L4

## 0 引 言

随着微电子技术和计算机技术的发展, 嵌入式技术得到广阔的发展空间, 特别是进入 21 世纪后, 嵌入式技术的发展和普及更加引人注目, 已经逐渐渗透到人们生活的各个方面。在通信领域, 众多网络设备如 WLAN、ADSL 等, 以及消费电子领域, 众多数码产品如 PDA、数码相机、MP3 播放器等都离不开嵌入式技术的支持。嵌入式系统具有资源相对有限、专用性强、高实时性<sup>[1]</sup>等特点, 因此应用于嵌入式系统的操作系统必须满足系统内核小、支持实时多任务、具有存储区保护功能以及可裁剪等要求, 微内核操作系统是能够

满足上述需求的技术途径之一。

微内核操作系统<sup>[2-3]</sup>结构是 20 世纪 80 年代后期发展起来的。为了提高操作系统的正确性、灵活性、易维护性和可扩充性, 在进行操作系统结构设计时大多采用基于客户端/服务器 (Client/Server) 的结构模式, 将操作系统划分为两大部分: 微内核和多个服务器。微内核并非是一个完整的操作系统, 而只是操作系统中最基本的核心部分, 它通常用于实现与硬件紧密相关的处理、实现一些基本的功能、负责客户和服务器之间的通信, 这样就可以确保把操作系统内核做得很小, 易于测试, 从而容易保证其正确性。操作系统的绝大

收稿日期: 2013-06-09

修回日期: 2013-09-18

网络出版时间: 2014-01-28

基金项目: 航空科学基金项目 (20111953016)

作者简介: 李 鹏 (1977-), 男, 高级工程师, 研究方向为嵌入式计算技术; 张 凡, 副教授, 研究方向为嵌入式计算技术、形式化方法。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140128.1149.044.html>

部分功能都放在微内核外面的一组服务器(进程)中实现,所有服务器都是运行在用户态,服务器与服务器之间采用的是消息传递通信机制,因此,当某个服务器出现故障时,不会影响内核,也不会影响其他服务器。这种设计思想使得基于微内核的操作系统具有高度的可重构性、稳定性和可靠性。正是由于微内核相比宏内核具有诸多优势,微内核操作系统在近三十年的时间里得到快速发展,已经被广泛应用于工业界,尤其在嵌入式系统领域微内核操作系统有着巨大的应用前景。

## 1 微内核操作系统发展简介

微内核是基于与宏内核不同的设计思路发展起来的,首个微内核是由卡内基梅隆大学(CMU)开发的基于 IPC 系统的 Accent 内核。这个内核不兼容 Unix 系统,因此只是一个实验性质的内核。为了实现对 Unix 的兼容,CMU 又在 Accent 的基础上开发了 Mach<sup>[4]</sup>微内核,并形成了基于该内核的操作系统。Mach 继承了 Accent 的进程间通信(IPC)<sup>[5-6]</sup>机制,但它本质上更像 Unix。Mach 和 Unix 的主要不同是,将大量的内核代码从内核态移到用户空间实现,这就使得内核变得相当小,但同时也使得线程间通信变得更加频繁,而线程间进行大量的消息传递使得 Mach 的性能低下,这也是制约第一代微内核操作系统发展的主要原因。

20 世纪 90 年代后期,微内核迎来了其生命中的第二春。一些研究人员认真分析了微内核系统性能差的原因,指出其性能差并非根本由内在的因素造成,而是设计实现的失误。为证明其论点,他们设计并实现了几个性能远超第一代的微内核操作系统,把它们称为第二代微内核操作系统,其中的一个代表作品就是 L4<sup>[7-9]</sup>。在 L4 设计过程中,采取了一系列的创新性的技术来提高 IPC 的效率,而这些技术在其他的微内核,例如 Exokernel, MINIX 3 的设计中也被广泛的使用,极大地提高了微内核的效率,进而使得微内核能够在实际应用中广泛使用。

近年来,随着嵌入式系统的不断发展,系统安全显得越来越重要,一种基于安全的微内核操作系统概念被提出,澳大利亚研究组织 NICTA 在 L4 基础上进行设计实现,形成 seL4<sup>[10]</sup>。seL4 微内核主要专注于应用程序接口(API)的形式化描述和形式化验证、安全的内核资源管理方法,这就要求内核在设计时必须考虑形式化验证<sup>[11-12]</sup>的需求,目前此类微内核操作系统还处于研究阶段。

## 2 L4 微内核实现机制分析

L4 微内核是 Jochen Liedtke 设计的基于微内核架

构的操作系统内核。L4 微内核遵循精简的设计原则,仅包含线程管理、地址空间管理、线程间通信、中断处理等基本功能,提供 11 个核心系统调用供开发者使用。

### 2.1 线程管理

L4 微内核中有两类线程,即特权线程和普通线程。L4 微内核运行之后,首先会调用两个特权线程 sigma0 和 root-task,并将系统控制权交给它们。sigma0 线程负责对内存空间进行管理,例如分配内存,root-task 线程负责对线程进行管理,包括线程的创建、撤销等。与 sigma0 或 root-task 处于同一个地址空间的线程即为特权线程,而其他地址空间的线程则为普通线程。

L4 微内核使用线程 ID 来标识线程,每个线程都有一个全局 ID(Global ID)和一个本地 ID(Local ID)。在内核中每个线程的全局 ID 是唯一的,而不同地址空间中的线程的本地 ID 可能是相同的,因此本地 ID 只能在线程的地址空间内使用,而地址空间间线程的通信应使用线程的全局 ID。

L4 微内核的线程状态可以分为以下四种:就绪状态、运行状态、挂起状态、等待状态,它们之间的转换关系如图 1 所示。

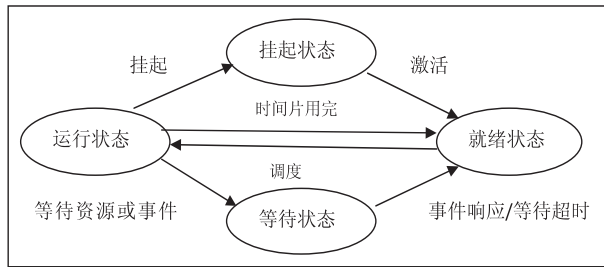


图 1 L4 微内核线程状态及转换关系

L4 微内核线程状态转换机制与传统的操作系统中线程状态转换机制基本相同,差别在于 L4 微内核线程调度可以按照其内部已有调度算法调度,也可以根据用户自定义调度算法进行调度,用户自定义调度通过调用线程管理原语在用户态实现。

### 2.2 地址空间管理

地址空间与线程是密切相关的,每个线程都运行在自己的地址空间内。L4 在初始化时创建一个特权线程 sigma0,它是一个运行于用户层的服务线程。内核将物理地址全部赋予该线程,以后对其他线程内存的分配管理都由 sigma0 线程负责。

不同地址空间的线程可以通过线程间通信机制来实现通信以及数据传输,同时 L4 微内核提供另一种机制来实现线程间的数据共享,即地址映射机制。地址映射机制提供地址空间管理原语来负责地址空间的映射,三个原语为:

(1)Grant:线程将本地地址空间指定区域授予其他地址空间的线程,从而使那些线程也可以访问该指定区域。Grant 之后,原线程将对该指定区域不再具有访问权。

(2)Map:与 Grant 相似,不同之处是共享之后原线程仍能够访问共享区域,即双方对共享区域共同拥有访问权。

(3)Unmap:通过该操作可以取消之前通过 Map 建立的共享关系。

基于这些原语,L4 微内核可以支持不同的用户线程以不同的策略来映射页面。

2.3 线程间通信机制

线程间通信是制约微内核操作系统性能的瓶颈,对其进行优化是学术界研究的热点。L4 微内核使用同步非缓冲通信机制,并通过多种方法使线程间通信性能获得大幅提高<sup>[8]</sup>,使用的主要方法有以下五点:

1)尽可能地用寄存器来进行消息传递,从而减少内存访问带来的开销。

2)通过聚合消息的方法将多条只包含发送操作的消息(最后一条可以包含接收操作)封装成单独的一条消息。

3)临时地址映射进行直接消息传输,将准备发送的数据的页地址直接映射到接收数据的线程所在的地址空间,使得接收线程能够对这些数据进行操作。

4)直接线程切换策略,发送数据的线程发送数据后可直接指定下一个调度的线程为接收线程,这时接收数据的线程可以直接被调度,以便接收数据。

5)懒惰调度,发送线程在发送数据之后,不会被直接阻塞,而是在就绪队列中等待。这时发送线程会被标记为等待数据接收状态,如果在规定的时间内数据被接收,并且接收线程返回数据已接收的消息,则将该标记取消;如果在规定的时间内没有完成数据接收,则将发送线程阻塞。

基于安全的原因,L4 微内核引入了时限机制。即如果超过一定时间对方没有响应则放弃本次线程间通信。系统服务选择无穷大作为其等待客户线程的时限,而客户线程则一般会选择 0 作为其等待服务线程的时限。

在 L4 微内核中有两种不同的线程间通信方式,即通用线程间通信和轻量级线程间通信,它们对用户是透明的,内核负责选用合适的线程间通信方式。在缺省的情况下,内核会调用通用线程间通信机制,只有在满足下面条件时才会调用轻量级线程间通信:

- 发送线程可向接收线程消息寄存器写消息;
- 接收线程的 ID 为本地 ID(与发送数据的线程在同一地址空间);

- 接收线程可以从消息寄存器中读取消息;
- 接收时限为无穷大;
- 通信过程不包括 Map 和 Grant 操作。

从上面的描述可以看出,LIPC 只能进行同一地址空间内的线程通信。

2.4 中断处理机制

L4 微内核把硬件中断当作是一些能够发送进程间通信消息给相关处理代码的线程,把中断服务程序当作是一些正在接收这些消息的线程。当一个硬件中断发生时,微内核会为这个中断产生一条消息,并把此消息发送给和此中断相关联的线程,该线程接收到这条 IPC 消息后处理这个硬件中断。内核仅负责产生中断消息,而不用涉及到具体的中断处理,从而使得中断处理的具体策略和内核隔离开来。这种处理方式使得中断处理程序可以在用户态运行。

2.5 系统调用

系统调用是 L4 微内核提供的系统服务接口,在程序设计中可以通过调用这些接口来实现对系统资源的访问,完成一些需要特权才能完成的操作。这种方式可以很好地保护系统资源,使其不会因为程序员的不当操作而被破坏。L4 微内核基本系统调用见表 1。

表 1 L4 微内核系统调用

接口名称	功能说明
SpaceControl	分配地址空间
ThreadControl	特权线程(例如,root server)可使用此系统调用进行线程控制(删除或创建线程、更改线程的全局 ID 等)
ProcessorControl	控制处理器的参数(例如,主频、电压等)
MemoryControl	设置 fpage(虚页)的页面属性
Ipc	用于进程间通讯和同步
Lipc	针对发送信息给本地地址空间的线程而进行优化的 IPC
Unmap	Unmap 指定的 fpage
ExchangeRegisters	交换或读取一个线程的 FLAGS、SP、IP 等信息
SystemClock	返回系统时钟
ThreadSwitch	请求的线程释放处理器,供另一个线程使用
Schedule	供调度器使用来定义优先级、时间片以及其他参数

从表 1 可以看出,L4 微内核提供的系统调用接口主要用于系统硬件资源的访问以及线程间的交互。这些系统调用接口都位于用户层,用户线程可以将它们当作普通的线程函数进行调用,完成其要进行的操作,例如当一个地址空间的用户线程想要同另一个地址空间的用户线程进行数据传输时,就可以通过调用 Lipc 或 Ipc 接口函数来完成。

3 基于 L4 微内核的应用开发

3.1 微内核的应用方式

根据不同的应用环境,微内核的应用分为三种不



同的方式<sup>[13]</sup>:

- 1)单服务器应用:微内核作为其他操作系统的载体,其他宏内核操作系统(Linux 等)运行在微内核之上。例如 I4Linux 等。
- 2)多服务器应用:执行操作系统功能的服务作为服务器运行在微内核之上,这种方式适合于分布式系统。例如 GunHurd 便是运行在 I4 之上的守护进程。
- 3)嵌入式系统的应用:微内核自身固有的特性使其能够很好地适应嵌入式系统应用,非常便于在不同的嵌入式平台上移植、裁减以及扩展<sup>[14]</sup>,例如 Vx-works。

文中着重讨论 I4 微内核在嵌入式系统中的应用。基于 I4 微内核操作系统的嵌入式系统软件结构框架如图 2 所示,用户程序和系统服务均处于用户层,利用 I4 微内核提供的系统调用(即系统服务进程接口)对用户线程进行初始化,例如分配地址空间、初始化线程控制块等,以及为用户线程提供服务,如用户线程间的数据传输、中断服务和线程间的切换等,用户程序通过调用系统服务进程接口完成特定任务。

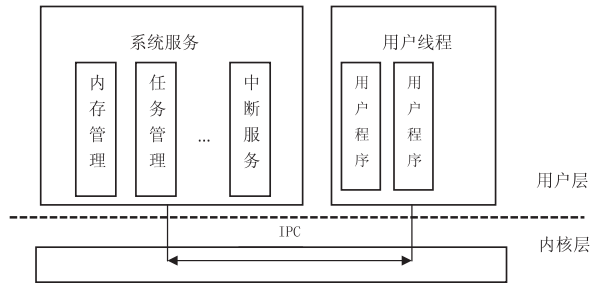


图 2 基于 I4 微内核操作系统的嵌入式系统软件结构框架图

将用户所编写的程序和 I4 微内核一起进行编译,然后引导加载到嵌入式系统中运行。I4 微内核支持不同的硬件结构,例如 ia32、amd64、powerpc 等。应用程序开发一般流程如图 3 所示。

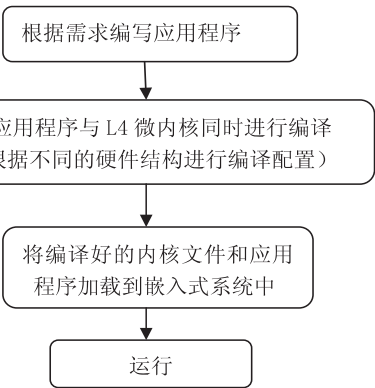


图 3 应用程序开发流程

3.2 应用实例

在嵌入式系统应用开发过程中,需要得到系统的一些硬件参数,这些参数有助于有效地避免软件移植

过程中出现的兼容性问题。通过设计一个获取运行环境参数的实例来具体描述上述应用开发过程。该实例主要实现如下 4 个功能:

- 1)打印 CPU 信息,如大端/小端,处理器个数,外频内频等;
- 2)打印内核信息,如时钟精度,内核名称/提供者,线程信息,地址空间信息等;
- 3)打印系统日期;
- 4)打印内存信息。

在实例中通过调用 I4 微内核中的函数来获得所需要的数据,例如调用 I4\_KernelInterface() 函数可以得到内核版本信息,代码片段如下:

```
I4_KernelInterfacePage_t * kip = (I4_KernelInterfacePage_t * )I4_KernelInterface();
I4_Word_t gen, year, month, day;
gen = ((I4_Word_t * )((I4_Word_t) kip + kip->KernelVerPtr))[1];
.....
```

利用 Qemu 作为 X86 体系结构的硬件仿真器,将开发好的应用软件与 I4 微内核进行编译,并使用 grub 在 Qemu 中加载 I4 微内核及应用程序,grub 引导程序中配置文件 menu.lst 片段如下:

```
Title L4
kernel /kickstart
module /x86-kernel
module /sigma0
module /Mytask
boot
.....
```

KickStart<sup>[5]</sup>是内核引导程序,负责加载 I4 微内核镜像文件 x86-kernel;sigma0 是 I4 微内核运行时的根线程,负责初始化地址空间(address space),加载用户应用线程 Mytask 等相关操作,Mytask 就是用户应用程序。运行结果部分截图如图 4。

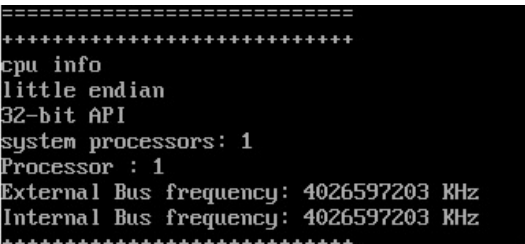


图 4 部分运行结果图

4 结束语

文中对 I4 微内核结构及实现机制进行了分析,进而介绍了 I4 微内核操作系统在嵌入式系统中应用的一般方法。随着嵌入式系统技术的不断发展,微内核 (下转第 37 页)

从实验结果可以看出,文中算法不仅能很好地保持图像的边缘特征,而且在去除噪声方面也能取得不错的分割效果。

5 结束语

文中提出了一种基于均值漂移和遗传算法相结合的新方法进行图像分割,详细介绍了均值漂移平滑图像的原理,通过调整  $h_s, h_r$  的值来平滑图像,然后通过遗传算法求出取得平滑图像的直方图最大熵时的阈值,通过与其他方法的比较可以得出,该方法对有噪声和无噪声的图像分割都能取得比较好的分割效果。

参考文献:

[1] 冈萨雷斯. 数字图像处理[M]. 第 2 版. 北京:电子工业出版社,2006.

[2] 何东健. 数字图像处理[M]. 西安:西安电子科技大学出版社,2003.

[3] 李俊山,李旭辉. 数字图像处理[M]. 北京:清华大学出版社,2007.

[4] 景晓军,蔡安妮,孙景鳌. 一种基于二维最大类间方差的图像分割算法[J]. 通讯学报,2001,22(4):71-76.

[5] 玄光男,程润伟. 遗传算法与工程优化[M]. 北京:清华大

(上接第 32 页)

操作系统将显得越来越重要,微内核操作系统已经成为嵌入式领域的研究热点。目前微内核的研究方向正逐渐从以前的如何提高内核效率转向内核的安全性,随着嵌入式系统在人们日常生活中逐渐普及,安全性将显得越来越重要,基于安全的微内核研究将会成为新的研究热点。

参考文献:

[1] 刘啸滨,郭 兵,沈 艳,等. 基于 ARM 处理器的嵌入式软件能耗统计模型[J]. 电子科技大学学报,2012,41(5):770-774.

[2] 邓 昀,程小辉,王新政. 微内核结构嵌入式实时操作系统的研究与设计[J]. 微电子学与计算机,2012,29(10):133-139.

[3] Liedtke J. Towards real microkernels[J]. Communications of the ACM,1996,39(9):70-77.

[4] Setapa S,Isa M A M,Abdullah N, et al. Trusted computing based microkernel[C]//Proc of international conference on computer application & industrial electronic. [s. l.]:[s. n. ], 2010:83-88.

[5] 陈少波. 一个微内核操作系统中进程管理的实现[J]. 制造

学出版社,2005.

[6] 张爱华,余胜生,周敬利. 一种二维直方图阈值化图像分割的后处理方法[J]. 华中科技大学学报:自然科学版,2002,30(10):59-61.

[7] Fukunaga K,Hostetler L D. The estimation of the gradient of a density function,with applications in pattern recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1975,21(1):32-40.

[8] Chen Yizong. Mean shift,mode seeking,and clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1995,17(8):790-799.

[9] Comaniciu D,Meer P. Mean shift:A robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2002,24(5):603-619.

[10] Pun T. A new method for grey-level picture thresholding using the entropy of the histogram[J]. Signal Processing,1980,2(3):223-237.

[11] Kapur J N,Sahoo P K,Wong A K C. A new method for grey-level picture thresholding using the entropy of the histogram [J]. Computer Vision,Graphics and Image Processing,1985,29(3):273-285.

[12] Holand J H. Adaptation in natural and artificial system[M]. Michigan:The University of Michigan Press,1975.

业自动化,2012,34(1):70-73.

[6] Jochen L. Improving IPC by kernel design[C]//Proc of 14th ACM symposium on operating system principles. Asheville, NC,USA:[s. n. ],1993:175-188.

[7] 王宽卿. 微内核进程间通信的研究[D]. 杭州:浙江大学,2010.

[8] Team L. L4 experimental kernel reference manual version X.2 [D]. Karlsruhe:Universität Karlsruhe,2011.

[9] L4ka Team. L4ka:Pistaciosource code 0.4[EB/OL]. [2012-04]. <http://www.l4ka.org/96.php>.

[10] Sewell T,Winwood S,Gammie P,et al. seL4 enforces integrity [C]//Proc of 2nd international conference on interactive theorem proving. Heidelberg, Berlin:Springer-Verlag,2011:325-340.

[11] 钱振江,刘 苇,黄 皓. 操作系统形式化设计与验证综述[J]. 计算机工程,2012,38(11):234-238.

[12] Bernard B,Heiser G. Correct,fast,maintainable:Choose any three! [C]//Proceedings of the Asia-Pacific workshop on systems. New York,NY,USA:ACM,2012.

[13] 邱 霆. 基于微内核的地址空间架构的研究[D]. 杭州:浙江工业大学,2008.

[14] 包海超,杨根庆,李华旺. 小卫星星载软件微内核的设计[J]. 计算机工程,2008,34(9):81-82.

作者：[李鹏](#)，[张凡](#)，[LI Peng](#)，[ZHANG Fan](#)  
作者单位：[李鹏, LI Peng\(中航工业西安航空计算技术研究所, 陕西 西安, 710068\)](#)，[张凡, ZHANG Fan\(西北工业大学 计算机学院, 陕西 西安, 710072\)](#)  
刊名：[计算机技术与发展](#)

---

英文刊名：[Computer Technology and Development](#)

---

年，卷(期)：2014(4)

本文链接：[http://d.wanfangdata.com.cn/Periodical\\_wjz201404007.aspx](http://d.wanfangdata.com.cn/Periodical_wjz201404007.aspx)